REMARKS

By the above amendment, Applicant has amended claim 20 to well define the scope and bounds. Applicant does not add "**involving more than 100 Boolean variables**" in the claims as in Amendment F(which was not entered) because the previous presented specification already defines a complex Boolean function in the claimed invention as a Boolean function **involving more than 100 Boolean variables,** which therefore cannot be processed using adjacency theorem or any other prior art referenced in the Final Action. Applicant also amended the specification to clarify why anything equivalent to a truth table is not practically useful for more than 100 variables. Applicant appreciates the good discussion in the Final Action on the interpretation of the complex Boolean function (as one with 6 or more variables) and on the adjacency theorem that is known to be dependent on representing the Boolean function as a truth table (or the equivalent). Applicant also appreciate the clear statement in the Advisory Action about adding "**involving more than 100 Boolean variables**" in claims.

With this amendment, it is clearer that the claimed invention has novel physical features that are not disclosed in the combination of references suggested in the Final Action, and the novel physical features are not obvious.

The Claim Rejection under 35 U.S.C. 112 Is Overcome

The Final Action rejected claim 20 under 35 U.S.C. 112, second paragraph, as being indefinite. Claim 20 has been amended to well define the scope and bounds. Applicant appreciates the good discussion in the Final Action on this issue.

The Claim Rejections On Okuzawa, etc. Is Overcome

The Final Action rejected all claims under 35 U.S.C. 103(a) over Okuzawa in view of Tucker and MPEP 2144.04(VI)(B) legal precedent for duplication, etc. The claims have been amended to define patentably over these references, and any combination thereof because now none of the limitations in the claims are disclosed by any of the references. Applicant requests reconsideration of this rejection for the following reasons:

(1) A complex Boolean function in the claimed invention are expressly defined in the specification as a Boolean function **involving more than 100 Boolean variables.** The specification clearly states "This invention relates to Boolean functions that involve more than 100 Boolean variables" in the first paragraph on page 2. The specification NEVER defines Boolean functions as involving any other number (e.g. 6 or more) of Boolean variables. Neither does the claims.

(2) **Adjacency theorem and any similar theory about simplification of complex Boolean functions are not practically useful** when there are more than 100 Boolean variables. By definition, adjacency theorem and any similar theory depend on inspecting the output value of the function at every

point in the input space. As clearly pointed out in the newly amended paragraph on page 3, it takes more than a trillion years to inspect all these output values regardless whether these values are stored exactly in a truth table. The amount of data is the same regardless whether these output values are listed as a truth table or anything equivalent to a truth table. It takes even longer time to compute all these output values. There are simply too many points in the input space. Therefore, being able to handle more than 100 Boolean variables is a "**new and unexpected result**" in the terms of **MPEP 2144.04(VI)(B)**. Adjacency theorem and all related methods are not "capable of being scaled up" in the terms of **MPEP 2144.04(IV)(A)**. Therefore, neither MPEP 2144.04(VI)(B) nor MPEP 2144.04(IV)(A) is applicable.

(3) **Tautology checking is a much simpler problem** than what adjacency theorem is meant to solve. It expects the output values at all points in the entire input space to be the same. There is no need to look at more output values if any 2 of these values are different! However, there is no previously known method for solving complex tautology checking problems, either. Therefore, it is a novel physical feature in the claimed invention that a **Boolean constant** is used in the comparison instead of the upper level logic or the lower level logic (represented as a logic circuit, a truth table or a Boolean expression) in Okuzawa. Because there are only 2 Boolean constants, it is possible to assume a default Boolean constant unless the other Boolean constant is given. It is not useful to assume any default logic circuit, truth table or Boolean expression.

(4) **It takes too long time to compare "a single input point at a time"** as suggested in the Final Action paragraph 37. There are simply too many input points when the input space involves more than 100 Boolean variables. As clearly pointed out in the newly amended paragraph on page 3, it takes more than a trillion years to do even the simplest operation on a single input point at a time because each row in a truth table (or the equivalent) represents an input point. This suggested approach in the Final Action might reduce the RAM requirement but it does not reduce the time requirement. Therefore, this kind of **"divide and conquer" in Tucker** is not practical for complex Boolean functions involving more than 100 Boolean variables.

(5) **It takes too much extra cost to allow "parallel processing"** as suggested in the Final Action paragraphs 37 and 106. As stated in Tucker page 2034, the second last paragraph, it is well known that "there is a cost associated with creating each task, whether due to communication over a network, increased contention for a shared memory, or context switching in the operating system." Only many trillion copies of the computing hardware can help, but the above cost slows it down very significantly for so many copies. There has been no known successful example of parallel processing with even merely several billion copies of the computing hardware. Therefore, **parallel processing** is not known to work for complex Boolean functions involving 100 Boolean variables.

(6) **There is no generally known method on how to choose "smaller subsets"** as suggested in the Final Action paragraphs 37 and 106. Because

an input space of more than 100 Boolean variables takes too long time to process either as a whole or a single point at a time with or without parallel processing, there is no known principle to show that any compromise (since small subsets are compromises between the entire space and individual points) can possibly be much more efficient. As described in the famous Amdahl's Law (which can be found in any parallel processing book), the maximum speedup of parallel processing is very low. Some reasons of Amdahl's Law are stated in Tucker page 2034, the second last paragraph. Therefore, there is no reasonable expectation to one killed in the art for using parallel processing to shorten the processing time from many trillion years to a reasonable amount of time.

(7) **It is invalid to compare "a single logical expression at a time (Tucker)"** as suggested in the Final Action paragraph 37. It only makes sense to compare 2 things against each other. It does not make sense to compare something against nothing.

(8) **Okuzawa** depends on representing Boolean functions as Binary Decision Diagrams as shown in its FIG. 1. Like Truth Tables, Binary Decision Diagrams are known not to scale well for complex Boolean functions of more than 100 Boolean variables. This poor scalability is shown in many of the references that were submitted with Information Disclosure Statement by Applicant. "Table 1: OBDD complexity for common function classes" of R.E. Bryant's "Symbolic Boolean Manipulation With Ordered Binary Diagrams", published in ACM Computing Survey, vol. 24, no. 3, September 1992,

showed examples of exponential complexities of Binary Decision Diagrams. J. R. Burch and V. Singhal said "It is well-known that this problem is coNP-hard" in the 5th and 6th lines under section title "1 Introduction" of their "Tight Integration of Combinational Verification Methods", published in IEEE/ACM International Conference on Computer Aided Design Digest of Technical Papers, November 1998. C.A.J. van Eijk said "As can be expected, there are types of circuits for which the BDDs become intractably large" on lines 4-5, page 134 in "A BDD Based Verification Method for Large Synthesized Circuits", published in INTEGRATION, The VLSI Journal, vol. 23, no 2, November 1997. J. Jain et al said "Due to the memory explosion problem, BDDs alone appear unsuitable for verifying large designs" immediately under the section title "4 Conclusion" on page 452 of their "A Survey of Techniques for Formal Verification of Combinational Circuits", published in Proceedings of IEEE International Conference on Computer Design: VLSI in Computers and Processors, October 1997. These references all discussed methods (many of which can be classified as "divide and conquer") to improve in special contexts, but they did not disclose any method similar to the one described in this application or any context used in this application.

(9) Because Okuzawa depends on Binary Decision Diagrams which are known not capable of being scaled up to 100 Boolean variables, **Okuzawa FIG 1 "LOGIC CIRCUIT", "BOOLEAN EXPRESSION" and "TRUTH TABLE"** involve less than 100 Boolean variables. Therefore they disclose neither "complex Boolean function involving more than 100 Boolean variables" *nor*

"given subset of the input space" (nor any other limitations/elements) in the rejected claims.

(10) Because Okuzawa depends on Binary Decision Diagrams which are known not capable of being scaled up to 100 Boolean variables, **Okuzawa FIG 1 "COMPARISON" and "SIMPLIFICATION"** involve less than 100 variables. Therefore, they do not disclose any limitations/elements in the rejected claims.

(11) **A key in handling more than 100 Boolean variables is to avoid repeating identical operations.** Parallel processing and "divide and conquer" only shift the heavy consumption of one resource to the heavy consumption of another resource, and such trade off between resources does not help as much as needed for 100 Boolean variables. The calculation of the output values at a point in the input space can involve many same operations as that at another point, especially if the 2 points are not far away from each other, though it usually also has some different operations. It requires carefully distinguishing what operations can be shared (and what cannot) to avoid repeating identical operations when calculating the output values at numerous points. The efforts to distinguish them must cost much less than the possible savings from such efforts. The efficiency improvement has to be very dramatic to turn the consumption of very huge amount of resources into the consumption of reasonable amount of resources. Achieving such dramatic improvement (turning the impossible into the possible) is **the unexpected, new, superior, usual, critical and surprising result** from the claimed invention. **The need**

of such improvement is long-felt but unsolved because it helps determining whether a large VLSI design is logically correct.

Conclusion

For all of the above reasons, Applicant submits that the specification and claims are now in proper form, and that the claims all define patentably over the prior art. Therefore he submits that this application is now in condition for allowance, which action he respectfully solicits.

Conditional Request For Constructive Assistance

Applicant has amended the specification and claims of this application so that they are proper, definite, and define novel structure which is also unobvious. If, for any reason, this application is not believed to be in full condition for allowance, applicant respectfully requests the constructive assistance and suggestions of the Examiner pursuant to M.P.E.P. § 706.03(d) and § 707.07(j) in order that the undersigned can place this application to allowable condition as soon as possible and without the need for further proceedings.

Very respectfully,

Zhe Li, Applicant

1 Argent Drive, Poughkeepsie, NY 12603

Telephone (845) 298-8342; Fax (720) 533-1988